IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

ERICSSON INC. and TELEFONAKTIEBOLAGET LM ERICSSON,	)	**CIVIL ACTION NO. 2:15-cv-291-JRG-RSP**
	)	**JURY TRIAL DEMANDED**
Plaintiffs,	)	
v.	)	
APPLE INC.,	)	
Defendant.	)	

ERICSSON INC. AND TELEFONAKTIEBOLAGET LM ERICSSON'S OPENING CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Page

CASES

STATUTES

OTHER AUTHORITIES

I. INTRODUCTION

The six patents asserted in this case claim an innovative cell phone architecture that allows for more efficient development, customization, and upgradability of cellular phones. For each disputed claim term, Plaintiffs Ericsson Inc.'s and Telefonaktiebolaget LM Ericsson's (collectively, "Ericsson") proposed construction is consistent with the intrinsic record and should be adopted. Apple Inc.'s ("Apple") proposed constructions, on the other hand, ignore the intrinsic record, or attempt to construe the claims using words that appear nowhere in the intrinsic record. Apple also proposes constructions more limiting than any preferred embodiment. Indeed, Apple advances absolute constructions such as "must," "always," and "all," even when the preferred embodiment imposes no such limitations, and instead describes that the embodiment "may" have a certain feature or "typically" performs certain functions. For these reasons, Ericsson asks the Court to reject Apple's proposed constructions and adopt Ericsson's proposed constructions as set forth in Exhibit A of the Second Corrected Joint Claim Construction Statement (Dkt. 88).

II. BACKGROUND OF THE TECHNOLOGY

The inventions claimed in the asserted patents describe improvements in mobile phone architecture that reduce the cost of developing cell phones, and provide an expandable platform for developers, manufacturers and end-users. Five of the patents, all of which include Figure 1, are drawn towards aspects of the mobile terminal system and platform assembly. The sixth patent (the '592 patent) is a related patent directed towards a further improvement in which the mobile terminal's access and application subsystems are separate and independently scalable.

U.S. Patent No. 7,536,181 (the "'181 patent") describes a mobile terminal platform system with a structured architecture. As depicted in Figure 1, the platform system includes a software services component that delivers software functionality through a series of software

modules; the modules are associated with corresponding hardware components, and are arranged in functional "stacks" with clearly defined functionality.[1] A middleware services layer provides an interface to access the software services, and thus the hardware components.[2]

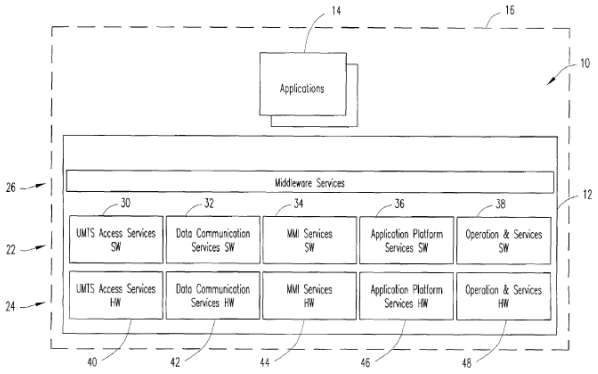

FIG. 1

The platform system includes application software that communicates with the underlying software and hardware via the interfaces in the middleware services layer.[3] The platform assembly "is adapted to be designed, implemented (assembled) and tested as a complete, enclosed unit separate from the application software."[4] This allows application developers to utilize the phone's underlying functionality via the middleware without being affected by any internal changes to the underlying functionality.[5] And, because application source code is not directly tied to specific software and hardware components, the arrangement allows for separate but parallel development of the applications and the mobile terminal platform assembly, and thus more efficient mobile phone development.

U.S. Patent No. 7,415,270 (the "'270 patent") further describes the middleware layer, and its use for development and addition of application software. The middleware services layer contains a number of application programming interfaces (APIs).[6] These APIs can be used to load, to install, and to run application software in a mobile terminal to complete the overall

---

[1] Ex. 3, '181 Patent at 3:65-4:2.
[2] Ex. 3, '181 Patent at 3:8-20, Fig. 1.
[3] Ex. 3, '181 Patent at 3:50-59.
[4] Ex. 3, '181 Patent at 3:50-53.
[5] Ex. 3, '181 Patent at 6:19-22.
[6] Ex. 4, '270 Patent at 2:34-43.

system.[7] The application software interacts with the APIs, which in turn provide access to the functionality offered by the underlying software services and hardware components. Using the inventions of the '270 patent, application developers do not need to know the specific details concerning the underlying services or hardware components when developing their applications.

U.S. Patent No. 7,149,510 (the "'510 patent") patent is directed to security aspects of the middleware services, and controlling access to services provided by the platform assembly. When an application requests access to a system service (e.g., the camera or location services), the request is intercepted and checked against the security policies to determine whether the request is authorized.[8] The ability to intercept software services requests ensures that only authorized applications are granted access.

U.S. Patent No. 8,079,015 (the "'015 patent") describes organizing software modules in mobile terminals into a plurality of horizontal layers.[9] These layers are ordered; higher-layered modules provide higher-level services than lower-layered modules.[10] The software modules in these layers communicate pursuant to a set of dependency rules.[11] The use of software layers and defined dependency rules makes it easier to compartmentalize and configure software functionality, which aids in application development and backward compatibility.

U.S. Patent No. 7,286,823 (the "'823 patent") is directed towards multimedia aspects of mobile terminals. The '823 patent allows easy development of applications that utilize multimedia functionality, such as video, music, and gaming applications.[12] The multimedia

---

[7] Ex. 4, '270 Patent at 2:18-23, 2:34-43.
[8] Ex. 6, '510 Patent at 7:42-52, 8:3-5, 8:11-14, 8:34-41.
[9] Ex. 1, '015 Patent at 2:3-7.
[10] Ex. 1, '015 Patent at 2:17-20.
[11] Ex. 1, '015 Patent at 2:22-25.
[12] Ex. 5, '823 Patent at 2:52-60.

applications utilize the middleware services layer to invoke the underlying functionality necessary to play video and audio content.[13] This underlying functionality includes the user interface, video codecs, audio functionality, and hardware components, among others, to play video and audio content on the mobile phone.[14]

U.S. Patent No. 7,707,592 (the "'592 patent") improves upon the platform architecture so that the access subsystem, which is responsible for cellular and data communications, and the application subsystem, which is typically responsible for user-application features, are independently scalable.[15] As an example, the cellular telecommunications hardware and software may be changed (i.e., differing or improved cellular technology) without affecting the application subsystem. Each subsystem is configured to run on a separate clock based on the needs of the subsystem — the access subsystem, for example, often requires a more accurate clock than the application subsystem to meet exacting cellular air-interface requirements.[16]

III. LEGAL PRINCIPLES

The principles of claim construction are well established. Claim terms are to be given their "ordinary and customary meaning," as determined by "a person of ordinary skill in the art in question at the time of the invention."[17] When construing the claims, the Court first considers intrinsic evidence, including the claims, the specification, and the prosecution history.[18] But the Court should not go so far as to write the specification into the claims.[19] A claim construction

---

[13] Ex. 5, '823 Patent at 6:42-50.
[14] Ex. 5, '823 Patent at 7:11-17.
[15] Ex. 2, '592 Patent at 2:13-31.
[16] Ex. 2, '592 Patent at 6:12-14, 9:55-60.
[17] *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2005) (en banc).
[18] *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc), *aff'd*, 517 U.S. 370 (1996); *see also Phillips*, 415 F.3d at 1315-17.
[19] *Id.* at 1322.

that "excludes the preferred embodiment is rarely, if ever, correct."[20] Conversely, the Federal Circuit "has expressly rejected the contention that if a patent describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment."[21]

A. Means-Plus-Function

In evaluating whether or not 35 U.S.C. § 112(6) applies to a particular claim term, "[t]he standard is whether the words of the claim are understood by persons of ordinary skill in the art to have a sufficiently definite meaning as the name for structure."[22] It remains settled law that "the failure to use the word 'means' . . . creates a rebuttable presumption . . . that § 112, para. 6 does not apply."[23] Sufficient structure may exist to avoid § 112(6) where a structural definition is provided in the specification or is known in the art.[24] Furthermore, when a claim element specifies a function and "recites sufficient structure or material for performing that function, § 112, P 6 does not apply."[25] If § 112(6) applies to a claim, the court must first identify the claimed function, and then it must identify what structure corresponds to that function.[26]

A. Indefiniteness

A party seeking to invalidate a patent bears the heavy burden of proving invalidity by clear and convincing evidence.[27] The Supreme Court has "read § 112, ¶ 2 to require that a

---

[20] *Adams Respiratory Therapeutics, Inc. v. Perrigo Co.*, 616 F.3d 1283, 1290 (Fed. Cir. 2010) (internal quotations and citations omitted).
[21] *Hill-Rom Servs. v. Stryker Corp.*, 755 F.3d 1367, 1371-1372 (Fed. Cir. 2014) (citation omitted) (collecting cases).
[22] *Williamson v. Citrix Online LLC*, 792 F.3d 1339, 1349-1350 (Fed. Cir. 2015).
[23] *Id.* at 1348-9. *Williamson* left intact the presumption as applied before the *Lighting World Inc. v. Birchwood Lighting, Inc.*, 382 F.3d 1354 (Fed. Cir. 2004) decision in September 2004.
[24] *See Apple Inc. v. Motorola, Inc.*, 757 F.3d 1286, 1299 (Fed. Cir. 2014) ("heuristics" term described sufficient input and output to avoid § 112(6)).
[25] *Allen Eng'g Corp. v. Bartell Indus.*, 299 F.3d 1336, 1347 (Fed. Cir. 2002).
[26] *Williamson*, 792 F.3d at 1351-1352.
[27] 35 U.S.C. § 282; *Microsoft Corp. v. i4i Ltd. P'ship*, 131 S. Ct. 2238, 2246 (2011).

patent's claims, viewed in light of the specification and prosecution history, inform those skilled in the art about the scope of the invention with reasonable certainty."[28] A means-plus-function term is definite if one of skill in the art would understand the specification to disclose structure, material, or acts that perform the recited function.[29]

IV. ARGUMENT

A. Term 1 – "defined dependency rules"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary. In the alternative, this term should be construed in accordance with its plain and ordinary meaning, i.e., "rules for communication between software modules."	"rules dictating how software modules must communicate"

In reference to Figure 1 (and other figures depicting the various possible software modules in the claimed architecture), this term relates to the manner in which software modules communicate. This term is readily understandable and requires no construction. In fact, the claims themselves already describe the "defined dependency rules":

> [W]herein the set of defined dependency rules includes a rule that a software module in a software layer may only invoke functionality in an interface in its own Software Back Plane or in Software Back Planes of software layers below its own Software Back Plane, and a rule that a software module may never invoke functionality in an interface in a Software Back Plane of a software layer above its own software layer.[30]

To the extent the Court determines that a construction would be helpful, the term should be construed as "rules for communication between software modules."[31] This construction is consistent with the specification, which explains that the dependency rules are "rules by which

---

[28] *Nautilus, Inc. v. Biosig Instruments, Inc.*, 134 S. Ct. 2120, 2129 (2014).
[29] *Creative Integrated Sys. v. Nintendo of Am., Inc.*, 526 Fed. Appx. 927, 936 (Fed. Cir. 2013).
[30] *See, e.g.*, Ex. 1, '015 Patent at 8:63-9:4, 10:12-20.
[31] Declaration of Robert Iannucci in Support of Ericsson's Opening Claim Construction Brief ("Iannucci Decl.") ¶ 27.

software modules in the various layers *may* communicate."[32] Apple, in contrast, substitutes "may" from the specification and file history, with "must" in its proposed construction. [33] This overly restrictive construction is not supported in the intrinsic record and should be rejected.

B. Term 2 – "Software Back Plane"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary. In the alternative, this term should be construed in accordance with its plain and ordinary meaning, i.e., "one or more layer interfaces."	"a computer-executable entity containing a group of software connectors allowing calls between software modules"

This term requires no construction because it is readily understandable by one of ordinary skill in the art. The software back plane comprises the interface — in other words, interconnections — between one or more layers.[34] To the extent construction is necessary, the term should be construed in accordance with its plain and ordinary meaning as "one or more layer interfaces."[35] The specifications of the asserted patents support this construction:

> The various modules in each layer implement the interfaces per function category to the layer. *These interfaces are gathered together in a Software Back Plane (SwBP) for each layer* which are separated from the module implementation.[36]
>
> The separation of the *layer interface (the SwBP)* from the layer implementation (modules), simplifies the exchange of module implementations and designs inside a layer as the modules are encapsulated by the SwBP.[37]

Apple's proposal is not supported by the intrinsic evidence and will add confusion. Nowhere do the claims, specification, or prosecution history use the phrases "computer-

---

[32] Ex. 1, '015 Patent at 6:29-30 (emphasis added); Iannucci Decl. ¶ 29.
[33] Ex. 1, '015 Patent at 6:29-30 (emphasis added); *see also* Ex. 7, '015 File History at ERIC_291 EDTX00006344 ("However, nothing in the cited passages of Goldstein discuss, expressly or inherently, anything about … 'defined dependency rules' *that dictate how software modules in the multiple software layers can communicate with each other* ….") (emphasis added).
[34] Iannucci Decl. ¶ 32.
[35] Iannucci Decl. ¶¶ 33-34.
[36] Ex. 1, '015 Patent at 5:49-52 (emphasis added); Ex. 3, '181 Patent at 5:23-26, 5:32-33.
[37] Ex. 1, '015 Patent at 6:45-48 (emphasis added); Ex. 3, '181 Patent at 5:23-26.

executable entity" or "group of software connectors allowing calls." Apple's proposal does not make the disputed term any clearer. Moreover, there is no support for the limitation that the back plane is "executable." Instead, Apple is improperly importing limitations into claims.

C. Term 3 – "interface means"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary. This element is not governed by 35 U.S.C. § 112(6). Not indefinite. To the extent the Court finds this element to be governed by 35 U.S.C. § 112(6), Ericsson proposes the following: **Function:** "permitting communication among software modules in said plurality of software layers pursuant to a set of defined dependency rules." **Structure:** "One or more of SWBPs 130, 132, 134, 140, 136, 138, 222, 224, 226 and 228, as described in '015 Patent at Figs. 3-5 and 5:49-53; 5:57-6:11; 6:45-48; 7:35-37; claims 1 and 10, that enable or facilitate communication on CPU 50 that may comprise one or more processors such as microprocessors, micro programmable processor or DSPs."	This term should be governed by 35 U.S.C. § 112(6). The function is permitting communication among software modules in said plurality of software layers pursuant to a set of defined dependency rules. Indefinite. The '015 patent does not disclose a corresponding structure, particularly an algorithm by which a general purpose computer performs the recited function.

"Interface means" should be given its plain and ordinary meaning; the interface between various modules facilitates communication between the modules.[38] Although this term includes the word "means," it is not a means-plus-function limitation.[39] When a claim element specifies a function and "recites sufficient structure or material for performing that function, § 112, ¶ 6 does not apply."[40] The Federal Circuit has found that "means" terms do not fall under the ambit of section 112(6) if the claims include a detailed recitation of structure after the "means" language:

> Most of these putative means-plus-function limitations contain far too much structure to claim the benefit of § 112, paragraph 6. For example, the 'gearbox

[38] Iannucci Decl. ¶ 35.
[39] Iannucci Decl. ¶ 35.
[40] *Allen*, 299 F.3d at 1347.

> means for rotating said blade means' of claim 15 is further defined in the claim to comprise 'a pair of rotatable shafts projecting downwardly from said frame means and defining a biaxial plane.' Similarly, the 'flexible drive shaft means for actuating said gearbox means' is defined by claim 15 itself to comprise 'individual shaft sections axially linked together by friction disk means for facilitating bending.' Such detailed recitation of structure clearly removes these limitations from the ambit of § 112, paragraph 6.' Because "'[a]ll of these limitations recite precise structure well understood by those of skill in the art... the word 'means' in these limitations may be ignored.'"[41]

Similarly, the "interface means" limitation in the claims at issue contains too much structure to be considered a means-plus-function term.[42] First, the "interface means" term comprises "a Software Back Plane for each software layer."[43] Second, the structure for each Software Back Plane is described as having "interfaces with the at least one module in its respective software layer."[44] Finally, the described structure interacts according to defined dependency rules that are further specified in the body of the claim.[45] All of these claim limitations recite precise structure well understood by those of skill in the art.[46]

However, even if the Court decides to construe this term as a means-plus-function term, this term is not indefinite. A person of ordinary skill in the art would understand that the

---

[41] *Id*. at 1348 (internal citations omitted).
[42] The full "interface means" limitation is: "an interface means for permitting communication among software modules in said plurality of software layers pursuant to a set of defined dependency rules; wherein the interface means comprises a Software Back Plane for each software layer, each Software Back Plane having interfaces with the at least one module in its respective software layer, and wherein the set of defined dependency rules includes a rule that a software module in a software layer may only invoke functionality in an interface in its own Software Back Plane or in Software Back Planes of software layers below its own Software Back Plane, and a rule that a software module may never invoke functionality in an interface in a Software Back Plane of a software layer above its own software layer." Ex. 1, '015 Patent at 8:57-9:4, 10:6-20; Iannucci Decl. ¶ 36.
[43] Ex. 1, '015 Patent at 8:60-61, 10:9-10; Iannucci Decl. ¶ 36.
[44] Ex. 1, '015 Patent at 8:61-63, 10:10-12; Iannucci Decl. ¶ 36.
[45] Ex. 1, '015 Patent at 8:63-9:4, 10:12-20; Iannucci Decl. ¶ 36.
[46] Iannucci Decl. ¶ 36.

specification discloses a specific algorithm on a specific programmed processor that is adequate to achieve the claimed function of "permitting communication among software modules in said plurality of software layers pursuant to a set of defined dependency rules."[47] The specification discloses a CPU 50 that may comprise one or more processors such as microprocessors, micro programmable processors or DSPs as shown in Figure 2, which is the processor that performs the above function.[48] The CPU 50 is programmed to perform the algorithms described in the patent.[49] As shown in the specification, in order to allow communication among software modules, the specific software backplanes 130, 132, 134, 140, 136, 138, 222, 224, 226 and 228 are used.[50] These software backplanes communicate using defined rules:

- A software module in a layer may only invoke functionality in interfaces in its Own SwBP or in SwBPs in layers below its own layer.
- A software module may never invoke functionality in interfaces in a SwBP above its own layer, independent of the layer to which the module belongs.
- A software module may invoke functionality in interfaces in the SwBP of its own layer in the same functional stack.
- There are no limitations for the direction of channel events or data streams. They may go in any direction.[51]

Therefore, there is sufficient disclosure in the specification of the algorithm executed by the specific processor to accomplish the recited function.

D. Term 4 – "middleware services layer/component"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"interfaces isolating the software architecture from the applications and supporting multiple application execution environments"

[47] Iannucci Decl. ¶ 38.
[48] Ex. 1, '015 Patent at 4:19-25; Iannucci Decl. ¶ 38.
[49] Ex. 1, '015 Patent at 4:19-25; Iannucci Decl. ¶ 38.
[50] Ex. 1, '015 Patent at 5:49-53, 5:57-31, 7:35-37, Fig. 3, Fig. 5; Iannucci Decl. ¶ 37.
[51] Ex. 1, '015 Patent at 6:1-11; Iannucci Decl. ¶ 38.

This term is understandable and need not be construed. One of ordinary skill in the art would understand that the "middleware services layer" contains "at least one application programming interface (API) for loading, installing and running one or more applications [] in mobile terminal platform assembly."[52] The "middleware services layer" may also provide various other services for applications, one of which is to "isolate[] the mobile terminal platform assembly from applications using it."[53]

Apple's proposed construction improperly narrows the scope of this term by attempting to read limitations from the exemplary embodiments into the construction of this term.[54] First, the "isolating the software architecture from the applications" function is just one of many functions that may be performed by the middleware services layer that is described in the exemplary embodiment of the patent. The specifications of the asserted patents explain that the middleware services layer "isolates the mobile terminal platform assembly from the applications using it, and … provides various other services for the applications."[55]

Second, Apple's proposed requirement that the middleware support multiple application execution environments also seeks to read limitations from the exemplary embodiment into the claims. Indeed, the specification of the '270 patent explicitly states that FIG. 4, which shows support for plurality of application environments, "is a block diagram that schematically illustrates details of middleware services layer component 26 according *to an exemplary embodiment of the present invention*."[56] In fact, the claims of the '270 patent contemplate an environment that could only support one application execution environment. For example,

---

[52] Ex. 4, '270 Patent at 4:12-16; Iannucci Decl. ¶ 40.
[53] Ex. 4, '270 Patent at 4:16-18; Ex. 1, '015 Patent at 3:52-56 ; Iannucci Decl. ¶ 41.
[54] *See, e.g., In re Am. Acad. of Sci. Tech. Ctr.*, 367 F.3d 1359, 1369 (Fed. Cir. 2004).
[55] Ex. 4, '270 Patent at 4:16-18; Ex. 1, '015 Patent at 3:52-56; Iannucci Decl. ¶¶ 40-41.
[56] Ex. 4, '270 Patent at 6:17-19 (emphasis added).

independent claims 1 and 18 of the '270 patent make no mention of how many different application execution environments may be supported by the middleware services layer.[57] Subsequently, the dependent claim 19 adds that the "application software is loaded, installed and run in said mobile terminal platform assembly via one of a native API domain and a non-native API domain," thus confirming that the middleware services layer could support only a single (native or non-native) application execution environment.[58]

E. Term 5 – "mobile terminal"

Ericsson's Proposed Construction	Apple's Proposed Construction
"A portable device designed for use in a cellular network"	The preambles do not serve as limitations of the claims of the asserted patents. Nevertheless, "mobile terminal" has its plain and ordinary meaning.

"Mobile terminal" should be construed as "a portable device designed for use in a cellular network." The asserted claims containing this limitation are directed to portable devices in cellular networks, and should not be expanded to cover other devices such as computers or laptops that are not designed for the purpose of being used in a cellular network.[59]

The genesis of the claimed inventions related to the advent of new cellular technologies that enabled mobile phone functionality beyond the traditional use of such devices for making phone calls.[60] The specifications describe that when cellular telecommunications systems were introduced in the 1980s, "the mobile terminals were designed primarily to provide voice telephony services; i.e., to receive and transmit voice communications."[61] In later years, mobile phones were developed so that in addition to voice communication, they included the ability to

[57] Ex. 4, '270 Patent at claims 1 and 18; Iannucci Decl. ¶ 42.
[58] Ex. 4, '270 Patent at claim 19; Iannucci Decl. ¶ 42.
[59] Iannucci Decl. ¶ 43.
[60] *See, e.g.*, Ex. 3, '181 Patent at 1:33-39; Iannucci Decl. ¶ 45.
[61] *See, e.g.*, Ex. 3, '181 Patent at 1:21-26; Iannucci Decl. ¶ 44.

transfer user data not related to the voice telephone call.[62] When the applications were filed, next generation cellular systems were under development that would provide "streaming audio/video, positioning, video conferencing and many other capabilities in addition to voice communication."[63] This new data transfer capability "drastically increased functionality that [was] being included in cellular telecommunications systems[.]"[64] Such increased functionality brought a need to provide the ability for multiple users to execute standardized applications on the mobile phones using scalable platforms.[65] This was challenging because mobile phones, unlike computers, were "resource scarce" and "limited in size, memory and power."[66]

The specifications confirm that the patents were directed to cellular phones – not computers. The '015 patent discusses how the data communications services software stack provides "services relating to connecting the mobile terminal to a PC," services relating to "connecting a PC to the Internet via the mobile terminal," and services relating to connecting "a mobile terminal to another mobile terminal."[67] In another example, the '592 patent states that "datacom services may also be adapted to permit dial up networking from a laptop or PDA using any of a plurality of physical interfaces such as, for example, the BT interface 236 or the IrDA interface 238."[68] Thus, the applicants' use of the terms "PC," "computer," and "laptop," did not intend to place such devices under the umbrella of "mobile terminals."

The file history of the '015 patent further confirms that the inventions in the asserted patents are limited to mobile terminals and not computers. On December 3, 2008, in an office

---

[62] Iannucci Decl. ¶ 44.
[63] *See, e.g.*, Ex. 3, '181 Patent at 1:33-39; Iannucci Decl. ¶ 44.
[64] *See, e.g.*, Ex. 3, '181 Patent at 1:48-51; Iannucci Decl. ¶ 45.
[65] *See, e.g.*, Ex. 3, '181 Patent at 1:55-65; Iannucci Decl. ¶ 45.
[66] *See, e.g.*, Ex. 3, '181 Patent at 1:51-54; Iannucci Decl. ¶ 45.
[67] *See, e.g.*, Ex. 1, '015 Patent at 7:49-57; Iannucci Decl. ¶ 46.
[68] *See, e.g.*, Ex. 2, '592 Patent at 8:17-20; Iannucci Decl. ¶ 46.

action the examiner required the patentee to restrict the invention to one of the following:

I. Claims 8, 10-18, 20-25, drawn to a software architecture in a mobile terminal, classified in class 455, subclass 419

II. Claims 26, 28-30, drawn to a software architecture in a computer, classified in class 709, subclass 217.[69]

The examiner explained that "the method of organizing a software architecture in a computer does not have to occur in a mobile terminal. The subcombination has separate utility such as for use in a mobile terminal."[70] In response, the applicant elected Group I, confirming that the inventions are directed to mobile terminals and not computers.[71]

F. Term 6 – "application subsystem comprising hardware and software for providing user application services"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"partitioned hardware and software that support all user applications"

The "application subsystem comprising hardware and software for providing user application services" is a term that would be readily understood by a person having ordinary skill in the art at the time the invention was filed, and need not be construed.[72] The term is descriptive and will be clear to the jury. Consistent with the understanding of one of ordinary skill in the art and the plain and ordinary meaning of this term, the '592 Patent generally states that "[t]he application subsystem includes hardware and software for providing user application services."[73] The specification further explains that, "[t]he application subsystem 300 *typically* handles all application services that can be configured to a specific product; therefore, the

[69] Ex. 7, '015 Patent File History at ERIC_291 EDTX00006195.
[70] Ex. 7, '015 Patent File History at ERIC_291 EDTX00006195.
[71] Ex. 7, '015 Patent File History at ERIC_291 EDTX00006199.
[72] *See* Iannucci Decl. ¶¶ 48-53.
[73] Ex. 2, '592 Patent at 2:14-17; Iannucci Decl. ¶¶ 50-51.

application subsystem 300 scales with the type of services needed and the performance of the services supported."[74] This passage describes a flexible and scalable set of services for executing user applications. Apple, however, seeks to add improper limitations. In particular, Apple adds the requirement that the subsystem support *all* user applications even though the specification indicates that the "application subsystem" "typically" supports all applications. Apple also adds that the application subsystem be "partitioned," which does not appear in the specification. The addition of "all" and "partitioned" is improper, and should be rejected.

G. Term 7 – "stack"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"encapsulated collection of hardware and software for particular functions"

The term "stack" is a well-understood term in the art, and need not be construed.[75] The specification does not stray from the common understanding of the term "stack," and explains that the "mobile-terminal platform system 100 includes a plurality of service stacks 106-122, each of which includes integrated hardware and software components that provide various functionalities of the mobile-terminal platform system[.]"[76] Such stacks are depicted in Figure 1.

Apple's proposed definition is incorrect because it adds unwarranted limitations. The word "encapsulated" is not used once in the specification, nor were any stacks specifically defined as relating to "hardware and software for *particular* functions." Apple's definition merely adds limitations that will only serve to confuse the jury, and would not help a person having ordinary skill in the art understand the definition of the well-understood term of "stack."

---

[74] Ex. 2, '592 Patent at 10:21-25 (emphasis added); Iannucci Decl. ¶ 52.
[75] *See* Iannucci Decl. ¶¶ 54-56.
[76] Ex. 2, '592 Patent at 3:52-56; Iannucci Decl. ¶¶ 55-56.

H. Term 8 – "interface services stack"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"[stack] that carries all traffic, data, control between the access and the application subsystems"

The term "interface services stack" has a well-understood meaning to a person of ordinary skill in the art, and need not be construed.[77] The interface services stacks of the '592 patent are stacks (as depicted in Figure 1) that facilitate communications between the application subsystem and the application subsystem. As the specification generally explains:

> The [interface] (IF) service stack 114 corresponds to and communicates with the IF service stack 112. Data communications 128 and control 130 between the access subsystem 102 and the application subsystem 104 occur via the IF service stack 112 and the IF service stack 114.[78]

Apple's proposed construction is improper because it imposes the requirement that "all traffic data and control" between the access subsystem and application subsystem must be carried by the interface services stacks. There is no requirement in the patent, however, that "all" communications between the subsystems be carried by the interface services stack, much less that all "traffic, data and control" be so carried. Apple's construction should be rejected.

I. Term 9 – "access subsystem comprising hardware and software for providing connectivity services"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"partitioned hardware and software controlling all external connections"

The term "access subsystem comprising hardware and software for providing connectivity services" is readily understood and need not be construed.[79] As the patent generally explains, "[t]he access subsystem includes hardware and software for providing connectivity

---

[77] *See* Iannucci Decl. ¶¶ 57-60.
[78] Ex. 2, '592 Patent at 4:6-10 ; Iannucci Decl. ¶¶ 58-59.
[79] *See* Iannucci Decl. ¶¶ 61-63.

services."[80] Thus, the term should be given its plain and ordinary meaning.

Apple's proposed definition is improper because it reads limitations into the claims. Although the access subsystem is largely concerned with connectivity, at least some external functionality may be shared between both the application subsystem and the access subsystem.[81] Notably absent from any description of the access subsystem is a requirement that it must control *all* external connections, or that the access subsystem be "partitioned." At no point does the patentee specify that partitioning or that "all external connections" are limitations of the access subsystem. Apple's attempt to again import the restrictive limitation of "all" into their proposed definition runs against the flexibility meant to exist within the scalable architecture of the claimed invention, and should be rejected in favor of the plain and ordinary meaning.

J. Term 10 – "are independently scaled"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"have been independently optimized and integrated for different responsibilities"

The term "are independently scaled" is understandable.[82] The concept of scaling is explained in a number of ways throughout the specification, such as "enabl[ing] the configuration of services required for development of cost-centric and size-centric devices."[83] In all instances, the term is simple to understand, and does not require further construction.[84]

In contrast, Apple's proposed definition is incorrect for two reasons. First, Apple's change of "are" to "have been" is not supported by the intrinsic record. Second, Apple's

---

[80] Ex. 2, '592 Patent at Abstract; Iannucci Decl. ¶ 62.
[81] *See* Ex. 2, '592 Patent at 15:9-23 ("Various external interfaces are part of the application subsystem 404 and the access subsystem 402.").
[82] *See* Iannucci Decl. at ¶¶ 64-68.
[83] *See* Ex. 2, '592 Patent at 5:32-47; Iannucci Decl. ¶¶ 65-66.
[84] *See* Iannucci Decl. at ¶¶ 67-68.

definition singles out language from the file history in an attempt to overly restrict the definition of "scalability." In an amendment to the claims, the patentee explained, in part, that "[t]he functional split means the application subsystem and the access subsystem can be independently optimized and integrated for different responsibilities handled by the access subsystem and the application subsystem, respectively. This is what is meant by being independently scaled."[85] This explanation is taken out of context, and in fact describes a particular example. Immediately prior to this particular example of scaling, the patentee states that "[t]he functional split *may* also be viewed as a separation between real-time control handled by the access subsystem and high-performance execution handled by the application subsystem."[86] In other words, in the context of adjusting real-time control and high-performance execution, independent scalability *may* relate to "independent optimization and integration for different responsibilities."

The patentee's use of permissive language makes clear that this is only part of the concept of scaling. The patentee also explained that "increases or decreases, that is, scaling, in the capability of the application subsystem can be done without regard for the real-time needs of the access side…. Similarly, adding or removing air interfaces and datacom interfaces may change the access clock requirements (USB for example) but don't affect the application part."[87] Additionally, the patentee explained that type approval in the access subsystem can be done without changes to the application subsystem, which "allows simpler scaling of the application part after type approval is completed."[88] The amendment provides many more instances to describe the general concept of scaling, none of which would *only* be discovered or understood

[85] Ex. 8, '592 Patent File History at ERIC_291EDTX00005008.
[86] Ex. 8, '592 Patent File History at ERIC_291EDTX00005008.
[87] Ex. 8, '592 Patent File History at ERIC_291EDTX00005014.
[88] Ex. 8, '592 Patent File History at ERIC_291EDTX00005014.

by importing the phrase "optimized and integrated for different responsibilities."[89]

K. Term 11 – "middleware"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"interfaces isolating the application and access subsystems from user applications and supporting multiple application execution environments"

The term "middleware" would be readily understood by a person having ordinary skill in the art, and does not need to be construed.[90] The '592 Patent generally explains that "[t]he access middleware (OPA) 124 has an application programming interface and a method of communicating with the application programming interface. In contrast, the application middleware 126 has an application execution environment and an application programming interface (OPA)."[91] The specification further explains that "[t]he mobile-terminal platform system 100 *may be adapted* to offer and support a range of different application environments for development of applications on the application subsystem 104."[92] Based on the concepts explained in these exemplary embodiments, a person of ordinary skill in the art would understand the plain and ordinary meaning of the term, and would not need a construction.

Although the specification states that "the application middleware 126 has *an application execution environment*…,"[93] Apple seeks to redefine this term as "multiple application execution environments." Further the specification explains that the system "may be adapted to offer and support a range of different application environments for development of applications,"[94] but Apple improperly proposes a construction that definitively requires support for multiple

---

[89] Ex. 8, '592 Patent File History at ERIC_291EDTX00005014-5015.
[90] *See* Iannucci Decl. ¶¶ 39-40.
[91] Ex. 2, '592 Patent at 4:18-22; Iannucci Decl. ¶ 40.
[92] Ex. 2, '592 Patent at 5:14-17 (emphasis added).
[93] Ex. 2, '592 Patent at 4:20-22 (emphasis added).
[94] Ex. 2, '592 Patent at 5:14-17.

environments. This language does not rise to the level of a clear and unambiguous disclaimer of scope or redefinition of the term "middleware."[95]

L. Term 12 – "loading, installing and running" terms

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary. Not indefinite.	Indefinite

The variations of this term from the '181, '270 and '823 patents are not indefinite and require no construction because they would be understandable by one of ordinary skill in the art at the time of the inventions.[96] The asserted patents explain how the users have the ability to load, install, and run application software in the platform assembly in order to provide a complete system for a mobile phone. The specifications of the patents explain:

> The present invention recognizes that by combining a plurality of functionally complementary units of software and hardware into an assembly that includes an interface component for providing access to the software, a mobile terminal platform assembly can be provided which can be marketed to a plurality of users (the term "user" as used herein includes manufacturers, end users or other customers or users). The users can then load, install and run their own application software in the assembly to provide a complete platform system for a mobile terminal. With the present invention, the mobile terminal platform assembly and the application software may be developed separately and then later combined to provide a complete platform system.[97]

The application software is loaded, installed, and run via an application programming interface (API) that is part of the middleware services layer.[98] The specification explains that loading (and installing) an application can include any mechanism by which the application

---

[95] Ericsson incorporates its arguments against Apple's construction with respect to the "middleware services layer" herein by reference. *See supra* Part III.E.
[96] Iannucci Decl. ¶ 69.
[97] Ex. 3, '181 Patent at 2:29-42; *see also* Ex. 4, '270 Patent at 2:49-65; Ex. 5, '823 Patent at 1:62-2:13; Iannucci Decl. ¶ 70.
[98] Ex. 5, '823 Patent at 4:15-18; Ex. 3, '181 Patent at 2:21-23, 3:38-41; Ex. 4, '270 Patent at 4:12-15; Iannucci Decl. ¶71.

software can be combined with the software of the mobile multimedia engine via the interfaces in the middleware including, for example, downloading from the Internet.[99] After the software is loaded and installed, it is run via the interfaces in the middleware services layer.[100]

An exemplary embodiment of the '823 patent in Figure 4 is illustrative. It shows various application software 204 (and application objects 212-220) that are loaded, installed and run via OPA (Open Platform API) objects, which are part of the middleware.[101]

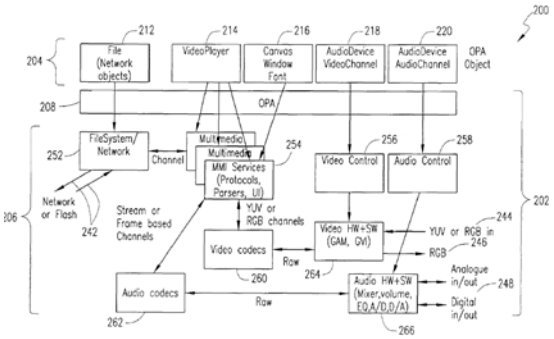

FIG. 4

Each multimedia application/object (e.g., Video Player Object 214) is first loaded and installed (e.g., downloaded from the internet and installed) and then when the application runs, it "accesses mobile multimedia engine 206 via application interface software (OPA) in middleware services layer 208."[102] A skilled person in the art would understand that these terms are not indefinite and should be given their plain and ordinary meaning.

M. Term 13 – "layer interfaces" terms

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary. Not indefinite.	Indefinite

These terms do not require construction. The terms "layer interfaces below its own layer," "layer interfaces above its own layer," and "layer interface in its own layer" in the '181 patent should be given their plain and ordinary meaning because the claims inform those skilled

---

[99] Ex. 5, '823 Patent at 2:47-51; *see also id*. at 8:51-53 ("Codecs may be downloaded via the Internet on demand or otherwise installed in the mobile multimedia engine."); Ex. 3, '181 Patent at 2:23-28; Ex. 4, '270 Patent at 2:38-43; Iannucci Decl. ¶ 71.
[100] Iannucci Decl. ¶ 71.
[101] Ex. 5, '823 Patent at Fig. 4; Iannucci Decl. ¶ 72.
[102] Ex. 5, '823 Patent at 6:48-50; Fig. 4; Iannucci Decl. ¶ 72.

in the art about the scope of the invention.[103] The "layer interfaces" (i.e., the software backplane) and how they are arranged are explicitly defined in the specification:

> As shown in FIG. 3, software services component 22, in addition to being organized into a plurality of vertical, functional software stacks as described above, is also arranged to define a plurality of horizontal layers . . . in which the *layers are arranged in descending order from a higher level service layer to a lower level service layer.*[104]

> The software itself is organized into a plurality of software modules, e.g. modules 102, 104, 106. In software services component 22, a single module can reside in only one vertical functional stack and in only one horizontal layer within that stack. *Each layer can contain from one to many modules, and all the modules in a particular layer and in a particular stack have the same level of abstraction. Communication among the various modules is accomplished via a Software Back Plane (SWBP) 112 subject to a set of basic rules for software module-to-module access.*[105]

> Apple claims that a lack of an explicit antecedent basis makes these terms indefinite.[106]

But the claim identifies multiple software functional units and the middleware services layer, which comprise the antecedent basis. A skilled person in the art would understand that these terms are not indefinite and should be given their plain and ordinary meaning.

N. Term 14 – "an application framework"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"a software framework for restricting access by non-native applications to platform services"

The term "application framework" of the '270 patent would be understood by a person having ordinary skill in the art, and does not need to be construed. The claims show that an application framework is generally used for "loading, installing and running" of the application

---

[103] Iannucci Decl. ¶ 74.
[104] Ex. 3, '181 Patent at 4:38-46 (emphasis added); Iannucci Decl. ¶¶ 75-76.
[105] Ex. 3, '181 Patent at 5:17-26 (emphasis added); Iannucci Decl. ¶ 77.
[106] *Energizer Holdings v. ITC*, 435 F.3d 1366, 1370-1371 (Fed. Cir. 2006) ("Despite the absence of explicit antecedent basis, 'If the scope of a claim would be reasonably ascertainable by those skilled in the art, then the claim is not indefinite.'" (quoting *Bose Corp. v. JBL, Inc.*, 274 F.3d 1354, 1359 (Fed. Cir. 2001)); s*ee also* MPEP § 2173.05(e) (9th ed., Mar. 2014).

software.[107] The prosecution history confirms the same:

> The middleware services layer in the present invention is further distinguished from Dehlinger by an 'application framework (204)' for loading, installing and running application software. This further limitation has been added to claims 1, 14 and 18.[108]

An exemplary embodiment in Figure 6 of the '270 patent shows that the application framework performs many functions related to loading, installing, and running of the application software.[109] For example, the application framework is responsible for granting access to the Open Platform API domain.[110] The application framework also performs registration, installation, starting, stopping, uninstalling and removal of all applications.[111]

The application framework supports both native and non-native applications. "Open Application Framework (OAF)," an embodiment, consists of two modules: "Secure Access Manager (SAM) module 220" and "an Application Manager (AM) module 222."[112] While the SAM module is "responsible for granting access to the Open Platform API domain 206 made by non-native applications," the AM module is "responsible for controlling the applications running in the non-native (Java) and *native execution environments*."[113] The application framework's ability to support native and/or non-native applications is further confirmed by the language of claim 19. "The method according to claim 18, wherein said application software is loaded, installed and run in said mobile terminal platform assembly via one of a native API domain and a

---

[107] Ex. 4, '270 Patent at claim 1; Iannucci Decl. ¶ 79.
[108] Ex. 9, '270 Patent File History at ERIC_291 EDTX00003094; Iannucci Decl. ¶ 80.
[109] Ex. 4, '270 Patent at 7:1-4 ("FIG. 6 is a block diagram that schematically illustrates . . . *another exemplary embodiment* of the present invention.")(emphasis added); Iannucci Decl. ¶ 80.
[110] Ex. 4, '270 Patent at 7:7-8; Iannucci Decl. ¶ 80.
[111] Ex. 4, '270 Patent at 7:21-23; Iannucci Decl. ¶ 80.
[112] Ex. 4, '270 Patent at 7:4-6; Iannucci Decl. ¶ 81.
[113] Ex. 4, '270 Patent at 7:6-9, 7:12-15 (emphasis added); Iannucci Decl. ¶ 81.

non-native API domain."[114]

Apple's attempt to limit the function of the application framework to "restricting access" should be rejected. The application framework is nowhere limited to only restricting access; it provides at least the functions of loading, installing, and running. Apple's limitation that the "application framework" relates only to non-native application is also improper. The application framework may handle either or both native and non-native applications.

O. Term 15 – "interface" / "interface component" / "middleware services layer"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary. Not indefinite.	Indefinite

Like the other "interface" related terms discussed above, these terms from the '823 patent are readily understandable to one or ordinary skill in the art and need no construction.

Apple asserts that the claims 1, 8, 15, and 19 (and their respective dependent claims) are indefinite because it is not clear whether "that isolates the mobile multimedia engine from the multimedia application software except via the at least one interface" modifies "middleware services layer," "interface," or "application program interface."[115] But one of ordinary skill would understand that the phrase "that isolates the mobile multimedia engine from the multimedia application software except via the at least one interface" applies to "middleware services layer."[116] The "middleware services layer" isolates the mobile multimedia engine from the multimedia applications via interfaces that are part of the "middleware service layer."[117] This is confirmed by the specification which explains the middleware services layer (1) "includes the at least one interface for loading installing and running the multimedia application

---

[114] Ex. 4, '270 Patent at claim 19.
[115] Ex. 10 (11/10/2015 Graubart letter to Chen) at 1-2.
[116] Ex. 5, '823 Patent at claim 18 does not contain the "via the at least one interface" language.
[117] Iannucci Decl. ¶ 84.

software in the mobile multimedia engine" and (2) "isolates the mobile multimedia engine from the application software except via the at least one interface."[118]

The file history of the '823 patent confirms that the middleware isolates the multimedia engine from the applications. The applicant stated that the middleware of the claimed invention "has the functionality of isolating a multimedia engine from a multimedia application software except via at least one interface."[119] The other asserted patents, which are incorporated by reference in the '823 patent, also confirm this understanding. For example, the specification of the '181 patent shows that the middleware services layer functions to isolate platform assembly from the applications.[120] These three terms are not indefinite, and need no construction.

P. Term 16 – "multimedia engine"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary. In the alternative, this term should be construed in accordance with its plain and ordinary meaning, i.e., "an engine that is capable of enabling multimedia applications"	"engine that performs all processing, handling, and transporting of multimedia data"

This term from the '823 patent requires no construction because this term is readily understandable. To the extent construction is necessary, the term "multimedia engine" should be construed in accordance with its plain and ordinary meaning as "an engine that is capable of enabling multimedia applications."[121] The specification states that a multimedia engine is a "mobile multimedia engine that is capable of enabling powerful multimedia applications."[122]

In contrast, Apple again reverts to importing an "all" limitation into the claim, though none is mandated by the intrinsic evidence. The language in an exemplary embodiment states

---

[118] Ex. 5, '823 Patent at 2:61-67, 4:14-20; Iannucci Decl. ¶ 84.
[119] Ex. 11, '823 Patent File History at ERIC_291 EDTX00001948.
[120] Ex. 3, '181 Patent at 6:5-9; *see also* claim 4.
[121] Iannucci Decl. ¶ 86.
[122] Ex. 5, '823 Patent at 6:7-8.

that "All processing, data handling and data transport is performed inside the mobile multimedia engine."[123] Contrary to Apple's proposal, the "processing" in the specification is not limited to "processing of multimedia data." Apple's proposal is overly narrow as it would potentially exclude a multitude of other functionality that the multimedia engine could perform (e.g., handling displays, handling cameras, handling network connections, providing persistent storage of multimedia content, audio/video playback, streaming audio/video, audio/video recording, video telephony, capability of adding decoders/encoders in runtime, etc.).[124] Finally, Apple is not attempting to make the disputed term more understandable to the jury. Instead, the addition of "processing," "handling," and "transporting" is likely to confuse the jury.

Q. Term 17 – "interface component"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"a software component between application domain software and the software services component and supporting multiple application execution environments"

No construction is necessary for the "interface component" term in the '510 patent.[125] The specification explains that "[t]he interface component has at least one interface for providing access to the software services component for enabling application domain software to be installed, loaded, and run in the platform."[126] And it states that the middleware services layer of the interface component 26 may support both a native and non-native environment, as shown in FIG. 4A.[127] There is no requirement, however, that both native and non-native environments or applications exist on the mobile terminal. The specification contemplates an environment that

[123] Ex. 5, '823 Patent at 7:25-27.
[124] *See generally*, Ex. 5, '823 Patent at 7:3-8:53; Iannucci Decl. ¶¶ 87-89.
[125] *See* Iannucci Decl. ¶¶ 91-93.
[126] Ex. 6, '510 Patent at 2:49-52; Iannucci Decl. ¶ 92.
[127] *See* Ex. 6, '510 Patent at 6:40-45, FIG. 4A.

does not have any non-native applications, leaving an interface component that supports only native applications.[128] Hence, the interface component can support either or both of the native and non-native applications, depending on the implementation. Apple's requirement that both types of environments must be supported should be rejected. A person having ordinary skill in the art at the time the '510 Patent was filed would understand the meaning of an "interface component," and no further construction is required.

R. Term 18 – "access controller"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"software securing the mobile terminal platform from application domain software to be installed in the platform"

No construction is necessary for the term "access controller." The '510 Patent itself is generally drawn towards a "[m]ethod and system for controlling access to a platform."[129] Access control, or more specifically an "access controller," has a meaning in the art. An "access controller" controls access to system services components from applications loaded, installed, and run in mobile terminal platforms.[130]

Apple's definition is incorrect because it unnecessarily complicates the claims, and renders them redundant. First, Apple's definition imports a limiting purpose for the "access controller" even though the claim already states its purpose:

> … [software securing the mobile terminal platform from application domain software to be installed in the platform] for controlling access to the software services component by a requesting application domain software via the at least one interface…[131]

---

[128] Ex. 6, '510 Patent at 7:9-11 ("[T]he Application Domain 500 need not necessarily hold any non-native applications.").
[129] Ex. 6, '510 Patent at Abstract.
[130] *See* Ex. 6, '510 Patent at 2:52-55 ("The system also includes an *access controller for controlling access* to the software services component by a requesting application domain software via the at least one interface.") (emphasis added); Iannucci Decl. ¶¶ 94-97.
[131] Ex. 6, '510 Patent at claim 1.

Apple's proposal also adds "securing" the mobile terminal platform, in addition to "controlling" access that is already included in the claim. And it also improperly limits the use of the platform to "application software to be installed in the platform." Nowhere in the specification is there language that expressly defines the "access controller" as software that merely concerns "application software to be installed in the platform."

S. Term 19 – "security access manager"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary.	"software that decides whether a request from [*non-native application domain software*] should be permitted"

No construction is necessary for the term "security access manager." The security access manager does what the name implies: manages secure access.[132] The security access manager can decide if access should or should not be granted. The '510 patent Abstract generally states that the invention includes an "interception module for receiving a request from the application domain software to access the software services component, and a security access manager for determining if the permission request should be granted."[133] Alternatively, the interception module at times can make an access decision locally, taking the "deciding" process away from the security access manager.[134] The specification explains that the application domain may hold either or both of native or non-native applications.[135] Regardless of the type of application, service requests are subject to access control as depicted in Figure 4B, which shows that the "Secure Access Manager" is functionally operative regardless of the type of application.

---

[132] *See* Iannucci Decl. ¶¶ 98-100.
[133] Ex. 6, '510 Patent at Abstract; Iannucci Decl. ¶ 99.
[134] *See* Ex. 6, '510 Patent at 8:35-41 ("The IM 223 grants or denies the request locally."); Iannucci Decl. ¶ 99.
[135] Ex. 6, '510 Patent at 7:5-10.

The interception module depicted above, which can use the "security access manager" to access permissions policies in certain embodiments, "might also be considered for native execution environments."[136] Indeed, the specification provides in its exemplary embodiment that "[a] service request is traffic between an application 250 (*See, e.g.*, FIGS. 6A B), *or any software in the application domain* 500, and the platform domain 502."[137] Thus, service requests, which may be intercepted by the interception module and subsequently checked by the "security access manager," contain functionality contrary to the limitation that Apple proposes.

T. Term 20 – "decision entity for determining if the request should be granted"

Ericsson's Proposed Construction	Apple's Proposed Construction
No construction necessary. This element is not governed by 35 U.S.C. § 112(6).† To the extent the Court finds this element to be governed by 35 U.S.C. § 112(6), Ericsson proposes the following: **Function:** "determining if the request should be granted." **Structure:** The security access manager (SAM) 518 or interception module (IM) 223 in the '510 patent.	This term should be governed by 35 U.S.C. § 112(6). The function is "determining if the request should be granted." The corresponding structure is: The security access manager (SAM) 518 in the '510 patent.

Within the context of the '510 Patent, a "decision entity for determining if the request should be granted" would be readily understood by a person having ordinary skill in the art. In general, a person of ordinary skill in the art would understand that a key concept of the invention is intercepting a system service request and determining whether or not that request is permitted.[138] Thus, no construction is necessary.

---

[136] Ex. 6, '510 Patent at 7:42-45.
[137] Ex. 6, '510 Patent at 7:56-60 (emphasis added).
[138] *See* Ex. 6, '510 Patent at 2:55-59 ("The access controller includes an interception module for receiving a request from the requesting application domain software to access the software services component and a decision entity for determining if the request should be granted."); Iannucci Decl. ¶¶ 101-104.

Apple argues that § 112(6) applies to this term. This term does not contain the word "means," which creates a presumption that § 112(6) does not apply.[139] Apple cannot overcome this presumption because a person of ordinary skill in the art could readily identify the structures described in the patent. The patent identifies two "entities" that may serve as the decision entity structure: the security access manager and/or the interception module.[140] In a first example, the '510 Patent explains that the interception module 223 can send the security access manager 518 a request for access to system services, at which point the security access manager can determine if access to such services should be granted.[141] In a second example, the interception module, instead of the security access manager, can make a decision locally to grant or deny access to system services.[142] These two examples make the structure of the decision entity clear to a person having ordinary skill in the art, thus negating the applicability of § 112(6).

In the event that the Court disagrees, and determines that § 112(6) does apply, Apple's definition excludes at least one described embodiment. As discussed above, decisions can be made either in the security access manager, or can be made locally in the interception module. The embodiment where the interception module makes a decision locally is improperly excluded from Apple's definition.[143] The structure must include at least both of these embodiments.

V. CONCLUSION

To the extent any construction is needed for the disputed terms, Ericsson's proposed constructions are consistent with the patent and should be adopted. Apple's proposals are based on misinterpretations of the law or the patent and should be rejected.

---

[139] *Williamson*, 792 F.3d at 1349.
[140] *See* Iannucci Decl. ¶ 103.
[141] *See* Ex. 6, '510 Patent at 8:7-13.
[142] *See* Ex. 6, '510 Patent at 8:35-41 ("The IM 223 grants or denies the request locally.").
[143] *Adams*, 616 F.3d at 1290.

Dated: December 15, 2015

Respectfully Submitted,

By: */s/ Denise M. De Mory*

Henry C. Bunsow (California State Bar #60707)
Lead Attorney
BUNSOW, DE MORY, SMITH & ALLISON LLP
351 California Street, Suite 200
San Francisco, California 94104
Telephone: (415) 426-4747
Facsimile: (415) 426-4744
Email: hbunsow@bdiplaw.com

Denise M. De Mory (California State Bar #168076)
Aaron R. Hand (California State Bar #245755)
BUNSOW, DE MORY, SMITH & ALLISON LLP
701 El Camino Real
Redwood City, California 94063
Telephone: (650) 351-7248
Facsimile: (650) 351-7253
Email: ddemory@bdiplaw.com
Email: ahand@bdiplaw.com

Mike McKool, Jr. (Texas State Bar #13732100)
Douglas A. Cawley (Texas State Bar #0403550)
Theodore Stevenson, III (Texas State Bar #19196650)
MCKOOL SMITH, P.C.
300 Crescent Court, Suite 1500
Dallas, Texas 75201
Telephone: (214) 978-4000
Facsimile: (214) 978-4044
Email: mmckool@mckoolsmith.com
Email: dcawley@mckoolsmith.com
Email: tstevenson@mckoolsmith.com

Samuel F. Baxter (Texas State Bar #01938000)
MCKOOL SMITH, P.C.
104 E. Houston Street, Suite 300
P.O. Box 0
Marshall, Texas 75670
Telephone: (903) 923-9000
Facsimile: (903) 923-9099
Email: sbaxter@mckoolsmith.com

Attorneys for Plaintiffs
ERICSSON INC. and
TELEFONAKTIEBOLAGET LM ERICSSON

CERTIFICATE OF SERVICE

I hereby certify that a copy of the foregoing document was filed electronically in compliance with Local Rule CV-5(a). Therefore, this document was served on all counsel who are deemed to have consented to electronic service. Local Rule CV-5(a)(3)(A).

Dated: December 15, 2015

/s/ Denise M. De Mory
Denise M. De Mory